Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844
Vol. V (2010), No. 5, pp. 968-975

Robust 2-DoF PID control for Congestion control of TCP/IP Networks

R. Vilanova, V. M. Alfaro

Ramón Vilanova
Department de Telecomunicació i Enginyeria de Sistemes
Universitat Autònoma de Barcelona
08193, Bellaterra, Spain,
E-mail: ramon.vilanova@uab.cat

Víctor M. Alfaro
Escuela de Ingeniería Eléctrica
Universidad de Costa Rica
San José, 11501-2060, Costa Rica.
E-mail: victor.alfaro@ucr.ac.cr

Abstract: This paper presents how Robust PID control can improve the performance of congesion control on TCP/IP networks. The proposed approach is compared with other control methods, such as PI control or RED/AQM, showing the advantages of the proposed technique.
Keywords: Congestion control, Active Queue Management, PID Control.

1 Introduction

Internet congestion control and congestion avoidance have been active research interests in the area of networking (see, for example [1] [2] [3]) during the last two decades. It has two components: (1) the end-to-end congestion control protocol, such as TCP [4], and (2) an active queue management (AQM) scheme implemented in routers. AQM signals congestion by discarding or marking packets. When congestion is detected by TCP, it will take actions to reduce the source sending rate. Normally, AQM objectives are: to stabilize the buffer queue length at a given target, thereby achieving predictable queueing delay, and to minimize the occurrences of queue overflow and underflow, thus reducing packet loss and maximizing link utilization. Thus, it is necessary to reduce as much as possible this problem. At present, there are methodologies to deal with this issue [5]: *congestion control* which is used after the network is overloaded and *congestion avoidance* which takes action before the problem appears. This paper deals with congestion control because it is where feedback control techniques can be openly and easily applied.

Recently several mathematical models of active queue management (AQM) schemes supporting transmission control protocol (TCP) flows in communication networks have been proposed [2] [3]. From these models a control theory-based approach can be used to analyze or to design AQM schemes. The more well known AQM scheme is probably RED [1]. RED can detect and respond to long-term traffic patterns, but it cannot detect congestion caused by short-term traffic load changes. In addition, it is well known that an appropriate tuning of RED parameters is not an easy task an may result in a non stabilizing controls scheme. This fact has motivated the research for alternative control approaches.

This paper presents the application of a Robust PID approach as an AQM controller. The controller can be easily tuned from the network parameters and it is compared with RED and the PI controller proposed in [3]. The performance under different load conditions shows the robustness and superiority of the presented approach. In addition the simple formulation of the PID controller also constitutes a motivation for implementation.

The rest of the paper is organized as follows. Next section presents the nonlinear model for a TCP router as well as the TCP control problem formulation. Section 3 reviews the RED and PI controller approaches for AQM control whereas section 4 presents the Robust 2-DoF PID approach. In section 5 a discussion and comparison is conducted. The paper ends with drawing some conclusions on the reported results.

2 AQM Router Dynamic Model and Control problem statement

In this section the dynamic nonlinear/linearized equations of TCP behavior developed in [3] are briefly reviewed as well as the purposes of AQM control stated.

2.1 Dynamic TCP Model

As in the literature, a network configuration consisting of a single congested router with a transmission capacity C is considered in this paper. TCP timeout mechanisms have been ignored for simplification. Using fluid-flow and stochastic differential equation analysis, the following coupled, nonlinear differential equations have been proposed as the dynamic model of the TCP behavior:

$$\begin{aligned} \dot{W}(t) &= \frac{1}{R(t)} - \frac{W(t)}{2}\frac{W(t-R(t-R(t)))}{R(t-R(t))}p(t-R(t)) \\ \dot{q}(t) &= \begin{cases} -C + \frac{N(t)}{R(t)}W(t) & q(t) > 0 \\ \max 0, -C + \frac{N(t)}{R(t)}W(t) & q(t) = 0 \end{cases} \end{aligned} \tag{2.1}$$

where

W	$\doteq$	average TCP window size (packets)
q	$\doteq$	average queue length (packets)
$R(t)$	$\doteq$	round-trip time$=\frac{q(t)}{C} + T_p$ (secs)
C	$\doteq$	link capacity
T_p	$\doteq$	propagation delay (secs)
N	$\doteq$	Number of active TCP sessions
p	$\doteq$	probability of packet mark

The queue length q, and window-size W, are positive bounded quantities; i.e., $q \in [0, \bar{q}]$ and $W \in [0, \bar{W}]$ where q and W denote buffer capacity and maximum window size respectively. In this formulation, the congestion window size W(t) is increased by one every round-trip time if no congestion is detected, and is halved upon a congestion detection. Moreover it has been assumed that the AQM scheme implemented at the router marks packets using Explicit Congestion Notification (ECN) to inform the TCP sources of impending congestion.

To linearize (2.1), it is assumed that the number of active TCP sessions and the link capacity are time-invariant, i.e., $N(t) \equiv N$ and $C(t) \equiv C$. In addition the dependence of the time delay argument $t - R$ on queue length q, is ignored and it is assumed to be fixed to $t - R_o$. Then, local linearization of (2.1) about the operating point results in the following equation:

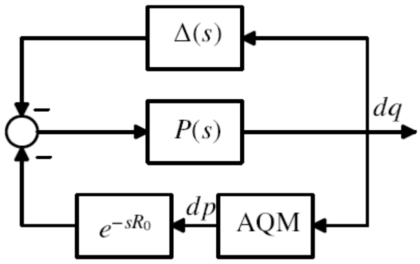

Figure 1: AQM Block Diagram of the linearized model along with High frequency uncertain dynamics

$$
\begin{aligned}
\delta \dot{W}(t) &= -\frac{N}{R_o^2 C}(\delta W(t) + \delta W(t - R_o)) \\
&- \frac{1}{R_o^2 C}(\delta q(t) - \delta q(t - R_o)) - \frac{R_o^2 C}{2N^2}\delta q(t - R_o) \\
\delta \dot{q}(t) &= \frac{N}{R_o}\delta W(t) - \frac{1}{R_o}\delta q(t)
\end{aligned}
\tag{2.2}
$$

where $\delta W(t) = W(t) - W_o$ and $\delta q(t) = q(t) - q_o$ are the incremental variables with respect to an operating point. The operating point for a desired equilibrium queue length q_o is given by:

$$R_o = \frac{q_o}{C} + T_p \quad W_o = \frac{R_o C}{N} \quad p_o = \frac{2}{W_o^2} \tag{2.3}$$

This leads to a low order nominal model of the network dynamics that is accurate at a particular operating point $(R_o, q_o, C_o, W_o, p_o)$ given by:

$$P(s) = P_o(s)e^{-R_o s} = \frac{C^2/(2N)}{(s + \frac{2N}{R_o^2 C})(s + \frac{1}{R_o})} \tag{2.4}$$

By modelling the high frequency dynamics using a block $\Delta(s)$ such that

$$\Delta(s) = \frac{2N^2}{R_o^3 C^3} s(1 - e^{-R_o s}) \tag{2.5}$$

This term represents the high frequency, necessarily parasitic, network uncertainty in the model. Computational experience has shown that this can adequately capture certain deviations from nominal network performance. These considerations lead to the generation of a simplified feedback control system as shown in Fig. (1)

2.2 AQM Control problem

The function of an AQM control law is to mark packets (with probability p) as a function of measured queue length q. Marking of packets is consecutively used by the sender to throttle the amount of data sent; if no marked packets are received the window size is increased. Upon reception of a marked packet the window size is halved. The principal performance objectives for an AQM control law are:

1. Efficient queue utilization, to avoid overflow or emptiness of the queue buffer.

2. Regulated queueing delay, to minimize (optimize) the time required for a data packet to be serviced by the routing queue. The queueing delay is equal to q/C.

3. Robustness, to maintain closed-loop performance in spite of plant uncertainties, N, R_o and C.

3 RED and PI Approaches to AQM

this section reviews two of the more well established approaches for AQM control, say RED and the PI approach presented in [3].

3.1 RED approach to AQM

Random Early Detection (known as RED) was presented by [1]. A RED gateway calculates the average queue size, using a low-pass filter with an exponential weighted moving average. The average queue size is compared to two thresholds (minimum and maximum). When the average queue size is less than the minimum threshold, no packets are marked. When the average queue size is greater than the maximum threshold, every arriving packet is marked. If marked packets are in fact dropped, or if all source nodes are cooperative, this ensures that the average queue size does not significantly exceed the maximum threshold. When the average queue size is between the minimum and the maximum threshold, each arriving packet is marked with probability p, where p is a function of the measured queue length q. Hollot et al. in [3] proposed the following transfer function model for the RED controller:

$$C_{red}(s) = \frac{KL_{red}}{s+K} = \frac{K_{red}}{s/k_{red}+1} \tag{3.1}$$

where

$$L_{red} = \frac{p_{max}}{(max_{th} - min_{th})} \quad K_{red} = \frac{R_o^3 C^2}{(2N)^2} \quad k_{red} = -C\ln(1-\alpha_{red}) \tag{3.2}$$

and where α_{red} is REDs queue-averaging weight. The corresponding controller block diagram is shown in figure (2).

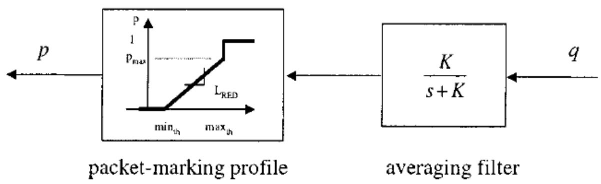

Figure 2: Block Diagram of RED as a cascade of low-pass filter and nonlinear gain element.

3.2 PI approach to AQM

According to [3] the transfer function of a PI controller can be written as:

$$C_{PI} = K_{PI}\frac{s/z+1}{s} \tag{3.3}$$

This controller is very well known by the control community. Its parameters can be tuned following methods proposed in the control literature. For example, [3] gave guidelines based on the Bode diagram tuning technique:

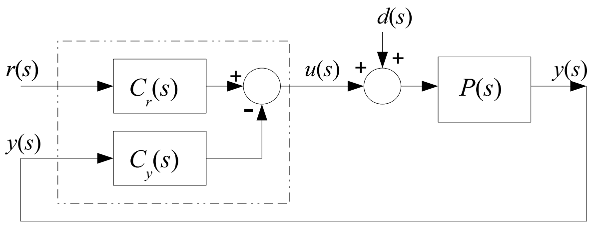

Figure 3: 2-DOF Control System

$$z = \frac{2N}{R_o^2 N} \quad K_{PI} = \omega_g z \left| \frac{j\omega_g + 1/R_o}{C^2/(2N)} \right| \quad \omega_g = \frac{\beta}{R_o} \tag{3.4}$$

where β determines the phase margin of the resulting nominal control system. The PI proposed in [3] is designed for a phase margin of about 30°.

4 Robust PID Control

This section briefly presents the approach for Analytical Robust Tuning (ART) of 2-DoF PID controllers recently presented in [6]. Consider the *Two-Degree-of-Freedom* (2-DOF) control system of Fig. 3, where $P(s)$ is the *controlled process* transfer function, $C_r(s)$ the *set-point controller* transfer function, $C_y(s)$ the *feedback controller* transfer function, and $r(s)$ the *set-point*, $d(s)$ the *load-disturbance*, and $y(s)$ the *controlled variable*.

The output of the 2-DOF controller is given by

$$u(s) = C_r(s)r(s) - C_y(s)y(s) \tag{4.1}$$

For a PID_2 [7] it is

$$u(s) = K_c\left(\beta + \frac{1}{T_i s}\right) r(s) - K_c\left(1 + \frac{1}{T_i s} + T_d s\right) y(s) \tag{4.2}$$

where K_c is the *controller gain*, T_i the *integral time constant*, T_d the *derivative time constant*, and β the *set-point weighting factor* $(0 \leq \beta \leq 1)$.

We will start right now with a Second-Order-Plus-Dead-Time (SOPDT) model of the form

$$P(s) = \frac{K_p e^{-L''s}}{(T''s+1)(aT''s+1)}, \quad \tau_o = \frac{L''}{T''} \tag{4.3}$$

The PID controller parameters are determined by the following equations for processes with parameters in the range $0.1 \leq \tau_o \leq 1.0$ and $0.15 \leq a \leq 1.0$.

$$\kappa_c = \frac{10\tau_i}{21\tau_c + 10\tau_o - 10\tau_i} \tag{4.4}$$

$$\tau_i = \frac{(21\tau_c + 10\tau_o)[(1+a)\tau_o + a] - \tau_c^2(\tau_c + 12\tau_o)}{10(1+a)\tau_o + 10a + 10\tau_o^2} \tag{4.5}$$

$$\tau_d = \frac{12\tau_c^2 + 10\tau_i\tau_o - (1+a)(21\tau_c + 10\tau_o - 10\tau_i)}{10\tau_i} \tag{4.6}$$

$$\beta = \min\left\{\frac{1}{K_c}, \frac{\tau_c T''}{T_i}, 1\right\} \tag{4.7}$$

The controller normalized parameters κ_c, τ_i and τ_d, and β depend on the model normalized dead-time τ_o and time constants ratio a, and on the design parameter τ_c. A minimum system robustness level is incorporated into the design process estimating a recommended maximum speed (τ_{cmin}) of the resulting closed-loop control system parameterized in terms of the maximum sensitivity function (M_s) by using

$$\begin{aligned} \tau_{cmin} &= k_{11}(M_s) + k_{12}(M_s)a^{k_{13}(M_s)} \\ & k_{11}(M_s) = 2.442 - 2.219M_s + 0.515{M_s}^2 \\ & k_{12}(M_s) = 10.518 - 8.990M_s + 2.203{M_s}^2 \\ & k_{13}(M_s) = 0.949 - 0.197M_s \end{aligned} \tag{4.8}$$

Combining the performance and robustness consideration above the design parameter may be selected in the range $\tau_{cmin} \leq \tau_c \leq 1.25 + 2.25a$. The range limits for the design parameter selection then combine the necessary restriction so that all controller parameters are positive and the accomplishment of a specified maximum sensitivity, with the necessity that the obtained response does not deviate too much away from the desired response, due of the dead-time approximation used in obtaining the tuning equations. For a more detailed presentation and discussion of the method please see [6].

5 Discussion

In order to illustrate the effectiveness of the Robust 2-DoF PID method, a numerical situation will be presented by taking the network simulation parameters of [3] where $q_o = 175$ packets, $T_p = 0.2$ seconds, $C = 3750$ packets/s (this corresponds to a 15MB/s link with an average packet size of 500 Bytes.). For a load of $N = 60$ TCP sessions we have $W_o = 15$ packets, $p_o = 0.008$, and $R_o = 0.246$. Therefore

$$P(s) = P_o(s)e^{-0.246} = \frac{1.17126\ 10^5}{(s+0.53)(s+4.1)}e^{-0.246} \quad \Delta(s) = 2.24\ 10^{-6}s(1-e^{-0.246s}) \tag{5.1}$$

The corresponding RED controller parameters are (see [3]) $K = 0.005$ and $L_{red} = 1.86\ 10^{-4}$ whereas those of the PI controller $K_{PI} = 9.64\ 10^{-6}$ and $z = 0.53$. On the other hand, for the application of the ART method the parameters of the SOPTD model are needed. These result to be $a = 0.1297$, $L'' = 0.2467$ and $T'' = 1.9014$. And the controller parameters depend upon the desired robustness level expressed in terms of te M_s level. A minimum robustness is assured by $M_s = 2.0$ whereas highly robust systems are designed with $M_s = 1.4$. Here we take an intermediate level with $M_s = 1.6$. The resulting 2-DoF PID controller parameters are: $K_c = 4.4241\ 10^{-5}$, $T_i = 2.1443$, $T_d = 0.3436$ and $\beta = 0.6172$. Figure (4) shows the performance of the three presented control strategies applied to the nonlinear system (2.1). As it can be seen the RED controller cannot reach the new references. This is an inherent drawback because it lacks the corresponding integrator. On the other hand both the PI and ART-PID controllers reach the desired targets. On the right part of figure (4) it can de observed that the ART-PID controller reaches the desired set-point faster and without any overshoot. In addition, a test under different load conditions has been performed. In this case the number of TCP sessions

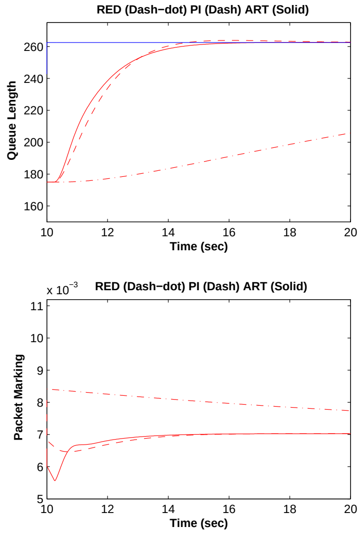

Figure 4: Changes on desired queue length. Comparison of PI, PID and RED performance. Plots on the right side show a zoomed version to better compare PI and PID

is a signal of the form $N(t) = N(1 + 0.01 * \sin(0.05t)) + \nu(t)$ being $\nu(t)$ a normally gaussian distributed random number of zero mean and variance $0.01 * N$. $\nu(t)$ is assumed to change with a sampling time of 5 sec. It can be seen that the ART-PID provides faster response to the load variation and faster recuperation of the desired queue length. Table (1) shows the mean value and standard deviation of the queue length error computed with respect to the desired target $q_o = 175 packets$. As a proof of performance, as we achieve a lower variation of the queue, a predictable performance will be expected. Therefore better QoS.

Table 1: Mean and standard Deviation of the queue length error

Controller	RED	PI	ART
mean	-0.43	-0.27	-0.1
std	10.9	9.2	5.0

6 Conclusions

In this paper the suitability of applying Robust PID controllers for the purpose of improving internet congestion control has been presented. The main advantage of the ART PID tuning is its one-parameter tuning. In addition this parameter is a direct specification of the desired robustness level. Therefore suitable for the sittuations with changes in load and system parameters. The performance has been compared to that of RED and a PI controller previously proposed in the literature.

Acknowledgment

This work has received financial support from the Spanish CICYT program under grant DPI2007-63356. Support from the Universidad de Costa Rica is greatly appreciated.

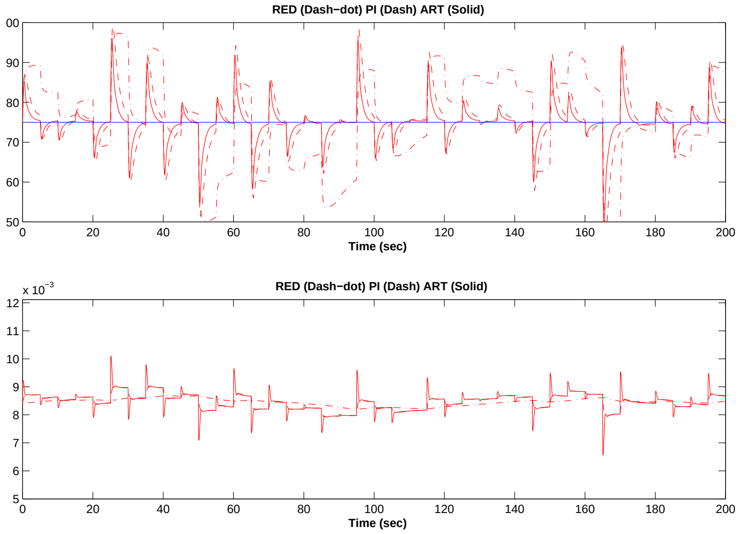

Figure 5: Regulation performance on a desired queue length of $q_o = 175 packets$ facing random variations of the number of TCP sessions

Bibliography

[1] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance.," *IEEE/ACM Transactions on Networking*, vol. 1, pp. 397–413, 1993.

[2] F. Kelly, "Mathematical modeling of the internet, in mathematics unlimited-2001 and beyond," *B. Enquist and W. Schmid, Eds, Berlin, Germany, Springer-Verlag.*, 2001.

[3] C. Hollot, V. Misra, D. Towsley, and W. Gong, "Analysis and design of controllers for AQM routers supporting TCP flows," *IEEE Trans. Automat. Contr.*, vol. 47, pp. 945–959, 2002.

[4] J. Postel, "Transmission control protocol," *RFC 793*, 1981.

[5] S. Ryu, C. Rump, and C. Qiao, "Advances in active queue management (AQM) based tcp congestion control," *Telecommunication Systems*, vol. 25, pp. 317–351, 2004.

[6] V. Alfaro, R. Vilanova, and O. Arrieta, "A single-parameter robust tuning approach for two-degree-of-freedom pid controllers," *The European Control Conference,Budapest, Hungary, August 23-26, 2009*, 2009.

[7] K. Åström and T. Hägglund, *Advanced PID Control.* ISA - The Instrumentation, Systems, and Automation Society, 2006.